CS 6230: High-Performance Computing and Parallelization
Course Syllabus

Instructor: Professor Mike Kirby

Course Information

Meeting Time: Tues/Thurs, 10:45 a.m. - 12:05 p.m.
Classroom: SCI Conference Room (WEB 3760)
Textbook: Parallel Programming (Second Edition) by Wilkinson and Allen
Web page: www.cs.utah.edu/~kirby/classes/cs6230
Office Hours: By appointment
Contact Information:

Office: WEB 3602
Phone: 585-3421
Email: kirby@cs.utah.edu

Teaching Assistant: Aravindan Thulasinathan (aravi [at] cs.utah.edu)
Office Hours: TBD

Course Description

This course is structured to train students to reason about the design and implementation of efficient parallel programs. The focus areas of the class will be on the modeling, implementation and evaluation of distributed, message-passing interface (MPI) based programs, shared-memory thread-based OpenMP programs, and hybrid (MPI/OpenMP) programs. Almost all examples will be aligned with numerical scientific computing. This course is appropriate for those that want transform serial algorithms into parallel algorithms, want to modify currently existing parallel algorithms, or want to write parallel algorithms from scratch.

Course Grading

Projects 1 and 2: 15% each
Final Project: 20%
Midterm: 25%
Final Exam: 25%

All three projects (Projects 1, 2 and the Final Project) will focus on designing, modeling, analyzing, implementing and evaluating parallel numerical algorithms. Projects will be graded based upon three items provided to the instructor: a type-written report, source-code, and a brief presentation. A handout will be provided to specify what the report and presentation should contain. Projects are to be done individually unless otherwise stated; if cheating is observed, students (all individuals involved) will receive a grade of 0% for the project assignment in question. Reports must be submitted in paper form at the beginning of class on the day in which they are due. Ten points per day will be deducted for late assignments (counting weekends). Late assignments may be presented in person or in PDF format by email. A tarball containing a report PDF

file, a presentation PDF file, source-code, and a README.txt file explaining how to compile and run the code must be provided by email to the instructor on the day that the project report is submitted (prior to 17.00hrs).

The Midterm and Final Exam will be closed book (and without notes). It is assumed that the student have read the assigned textbook chapters and supplementary papers (as questions will be taken from material in the text and papers not necessarily discussed in class). Students will have the entire class period to take an exam. Make-up exams will not be allowed except in university-approved extenuating circumstances (*i.e.* death in the family, illness documented by a physician, etc.).

How to Stay On Top Of Things

The best way to stay on top of things is to read the assigned textbook chapters and supplementary papers! The book and papers will, in most cases, be the guides to the lectures. Lecture slides will be posted; when possible they will be posted before class (although this is not always possible). The other way to stay on top of things is to frequently check the class schedule for new handouts, changes in course topics, etc.

Topics to be Discussed

Below is a tentative list of some of the topics/algorithms that will be discussed as part of the course. The specific topics to be discussed will be determined based upon faculty and student interests as the course evolves.

- Message-Passing Interface (MPI) based programming
- OpenMP thread-based programming
- Embarrassingly Parallel Programs
- Partitioning and Divide-and-Conquer Strategies
- Pipelined Computations
- Synchronous Communication
- Numerical Integration
- Sorting Algorithms
- Matrix/Vector and Matrix/Matrix Multiplication
- Solving of Linear Systems (Direct and Iterative Methods)
- Fourier Methods